**Федеральное агентство связи**

**Государственное бюджетное образовательное учреждение высшего профессионального образования ордена Трудового Красного Знамени**

**Московский технический университет связи и информатики**

Кафедра «Математической кибернетики и информационных технологий»

**Отчет по лабораторной работе №3**

по дисциплине «Теория языков программирования»

Выполнила студентка

группы БСТ1801

Куранова И.А.

Вариант 12

Москва 2019

1. **Цель работы:** освоить базовые навыки работы с функциями и файлами в Python.
2. **Задание:**

Реализовать задания в виде пользовательских функций.

Реализовать единое меню выбора соответствующих функций заданий №1-4 в виде:

0 – Выход из программы

1 – Название функции №1.

2 – Название функции №2.

3 – …

После выполнения каждой из функций запрашивать у пользователя «Вы хотите продолжить?» Если ответ «да» (yes, Y, 1), то снова выводите меню. Если ответ «нет» (no, N, 0), то завершите программу.

* 1. Посчитайте количество файлов в заданной пользователем директории (папке) и вывести на экран.
  2. Дан файл products.txt, в котором содержится информация о товарах в виде:

№;Наименование\_товара;Цена;Количество

1;Наушники;1549;53

2;Шагомер;1999;17

3;Диктофон;2099;6

...

Считайте информацию из файла в структуру: [[№, Название, Цена, Количество],[ [№, Название, Цена, Количество],[№, Название, Цена, Количество]] (список списков).

Выведите информацию о товарах дороже 1700, отсортировав их по названию.

* 1. Добавьте к задаче №2 пользовательский интерфейс:

По уменьшению цены всех товаров указанных пользователем номеров на введенное пользователем число.

* 1. Добавьте к пользовательскому интерфейсу из задачи №3 возможность сохранения новых данных обратно в файл. Предусмотрите возможность сохранения данных в отсортированном виде в отдельный файл по выбору пользователя.

1. **Ход работы:**
   1. Код программы задания:

**import** os  
f = open(**'Products.txt'**, **'r'**) *#открыть для чтения*product = []  
  
**for** line **in** f:  
 line = line.strip() *#удаление пробельных символов в начале и конце строки* line = line.split(**';'**) *#разбиение строки по разделению* product.append([line[0], line[1], line[2], line[3]]) *#добавляет эл-т в конец списка*maxi = int(len(product)) *#кол-во товаров  
  
  
#Задание 1 (Узнать кол-во файлов в папке)***def** count(): *#инструкция, определяющая функцию* path = input(**'Введите путь к папке: '**)  
 files = next(os.walk(path)) *#возврат файлов* print(**'Количество файлов в папке: '**, len(files))  
  
 print(**'---------------------------------------------------------------\n'  
 '1 - Повторить программу\n'  
 '2 - Меню\n'  
 '---------------------------------------------------------------\n'**)  
 cont = input(**'Выберете команду: '**)  
 **if** cont == **"1"**:  
 count()  
 **elif** cont == **"2"**:  
 menu()  
  
*#Задание 2 (Сортировка)***def** sor():  
 print(**'Сортировка по:\n'**,  
 **'1 - Названию (A-Z) - дороже 1700\n'**,  
 **'2 - Названию (Z-A) - дороже 1700'**)  
 a = input(**'Выберете команду: '**)  
 **if** a == **'1'**:  
 print(**' Таблица:'**)  
 product.sort(key=**lambda** line: str(line[1]))  
 **for** i **in** range(0, len(product)):  
 **if** int(product[i][2]) > 1700:  
 print(product[i])  
 **elif** a == **'2'**:  
 print(**' Таблица:'**)  
 product.sort(key=**lambda** line: str(line[1]), reverse=**True**)  
 **for** i **in** range(0, len(product)):  
 **if** int(product[i][2]) > 1700:  
 print(product[i])  
 **else**:  
 print(**'Неверная команда!'**)  
 sor()  
  
 print(**'---------------------------------------------------------------\n'  
 '1 - Повторить программу\n'  
 '2 - Меню\n'  
 '---------------------------------------------------------------\n'**)  
 cont = input(**'Выберете команду: '**)  
 **if** cont == **"1"**:  
 sor()  
 **elif** cont == **"2"**:  
 menu()  
 **return** product  
  
*#Задание 3 (Уменьшение цены)***def** price():  
 NUM = [] *#список* print(**' Таблица:'**)  
 **for** i **in** range(0, len(product)):  
 print(product[i])  
 print(**'0 - Далее'**)  
  
 work = **True  
 while** work **is True**:  
 num = input(**'Введите номер товара: '**)  
 **if** num.isdigit(): *#содержит ли строка только цифры?* num = int(num)  
 **if** num == 0:  
 work = **False  
 elif** num <= maxi:  
 NUM.append(num)  
 **else**:  
 print(**'Такого номера не существует! Максимальный номер:'**, maxi)  
 work = **True  
 else**:  
 print(**'Неверный символ! Повторите попытку.'**)  
 work = **True** minus = int(input(**'Введите значение, на которое нужно уменьшить цену: '**))  
 **for** i **in** range(0, len(product)):  
 **for** j **in** range(0, len(NUM)):  
 **if** int(product[i][0]) == int(NUM[j]): *#если номер товара равен введеному номеру* product[i][2] = int(product[i][2]) - minus  
  
 print(**'Изменения сохранены!\n'  
 ' Новая таблица: '**)  
 **for** i **in** range(0, len(product)):  
 print(product[i])  
  
 print(**'---------------------------------------------------------------\n'  
 '1 - Повторить программу\n'  
 '2 - Меню\n'  
 '---------------------------------------------------------------\n'**)  
 cont = input(**'Выберете команду: '**)  
 **if** cont == **"1"**:  
 price()  
 **elif** cont == **"2"**:  
 menu()  
 **return** product  
  
*#Задание 4 (Сохранение результата)***def** savage():  
 print(**' Где сохранить результат?\n'  
 '---------------------------------------------------------------\n'  
 '1 - Сохранить в этом документе\n'  
 '2 - Сохранить в новом документе\n'  
 '---------------------------------------------------------------\n'**)  
 a = input(**'Выберете команду: '**)  
  
 **if** a == **'1'**:  
 file = open(**'Products.txt'**, **'r+'**) *#открыт для чтения и записи* **for** i **in** range(0, len(product)):  
 **for** j **in** range(0, len(product[i])):  
 **if** 0 <= j <= 2:  
 file.write(str(product[i][j]) + **";"**) *#запись в файл* **else**:  
 file.write(str(product[i][j]))  
 file.write(**'\n'**)  
 print(**'Файл Products.txt успешно сохранен!'**)  
 file.close()  
  
 **if** a == **'2'**:  
 name = input(**'Введите название файла: '**)  
 new = open(name, **'w'**) *#открытие на запись, содержимое файла удаляется, если файла не существует, создается новый* **for** i **in** range(0, len(product)):  
 **for** j **in** range(0, len(product[i])):  
 **if** 0 <= j <= 2:  
 new.write(str(product[i][j]) + **";"**)  
 **else**:  
 new.write(str(product[i][j]))  
 new.write(**'\n'**)  
 print(**'Файл '**, name, **' успешно сохранен!'**)  
 new.close()  
  
 print(**'---------------------------------------------------------------\n'  
 '1 - Повторить программу\n'  
 '2 - Меню\n'  
 '---------------------------------------------------------------\n'**)  
 cont = input(**'Выберете команду: '**)  
 **if** cont == **"1"**:  
 savage()  
 **elif** cont == **"2"**:  
 menu()  
  
*#МЕНЮ***def** menu():  
 print(**'------------------------------МЕНЮ------------------------------\n'  
 '1 - Задание 1 - Открыть файл\n'  
 '2 - Задание 2 - Сортировка по названию\n'  
 '3 - Задание 3 - Уменьшение цены\n'  
 '4 - Задание 4 - Сохранить изменения\n'  
 '0 - Выход из программы\n'  
 '----------------------------------------------------------------\n'**)  
 a = input(**'Выберете команду: '**)  
  
 **if** a == **"1"**:  
 count()  
 **elif** a == **"2"**:  
 sor()  
 **elif** a == **"3"**:  
 price()  
 **elif** a == **"4"**:  
 savage()  
 **elif** a == **"0"**:  
 print(**' Пока! :)'**)  
 exit()  
 **else**:  
 print(**'Неверная команда!'**)  
 menu()  
  
  
menu()  
  
f.close()

* 1. Результат работы программы:

Скриншот работы программы меню (функции menu()) представлен на рисунке 1.

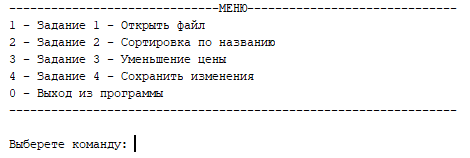


Рисунок 1 - Результат работы функции menu()

Скриншот работы программы первого задания (функции count()) представлен на Рисунке 2.

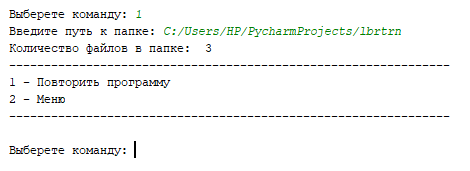


Рисунок 2 - Результат работы функции count()

Скриншот работы программы второго задания (функции sor()) представлен на Рисунке 3.

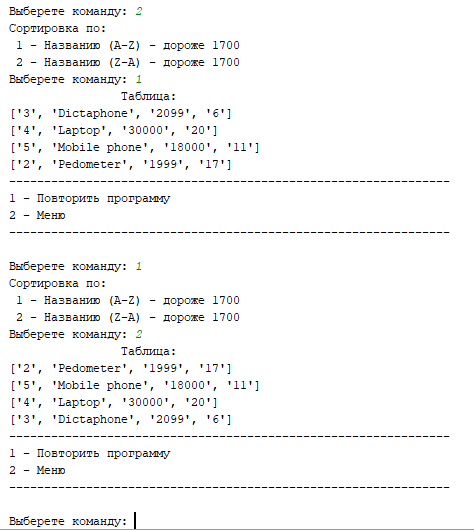


Рисунок 3 - Результат работы функции sor()

Скриншот работы программы третьего задания (функции price()) представлен на Рисунке 4.

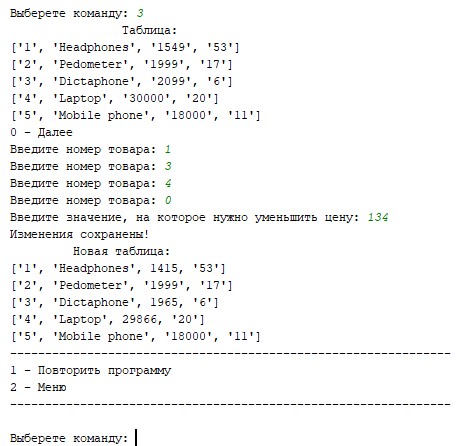


Рисунок 4 - Результат работы функции price()

Скриншот работы программы четвертого задания (функции savage()) представлен на Рисунке 5.

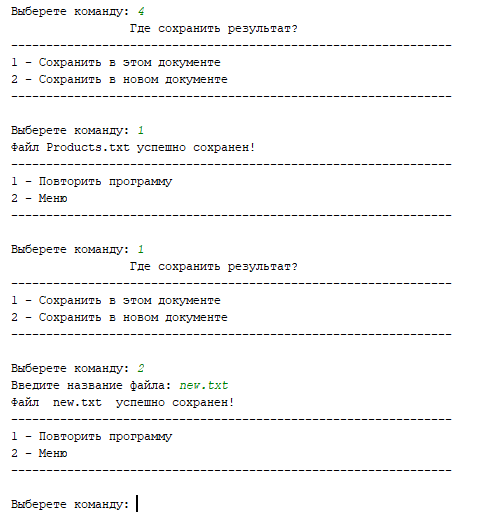


Рисунок 5 - Результат работы функции savage()

1. **Ссылка на репозиторий:**

<https://github.com/cryuso/ToLP_labs/tree/master/Lab3>

1. **Вывод:** Я освоила базовые навыки работы с функциями и файлами в Python.